Distributed Formation Control of Quadrotors under Limited Sensor Field of View*

Duarte Dias[1,2], Pedro U. Lima[2], Alcherio Martinoli[1]
[1]Distributed Intelligent Systems and Algorithms Laboratory, EPFL, Lausanne, Switzerland
[2]Institute for Systems and Robotics, IST, Universidade de Lisboa, Lisboa, Portugal
[1,2]duarte.dias@epfl.ch

ABSTRACT

This work tackles the problem of quadrotor formation control, using exclusively on-board resources. Local inter-robot localization systems are typically characterized by limited sensing capabilities, either in range or in the field of view. Most of the existing literature on the subject uses inter-robot communication to obtain the unavailable information, but problems such as communication delays and packet loss can seriously compromise the system stability, especially when the system shows fast dynamics such as that of quadrotors. This work focuses on the sensor field of view limitation: it proposes a formation control algorithm that extends the existing methods to allow each quadrotor to control the occupied area of its sensor field of view, while moving to the right place in the formation. This decreases the situations when necessary inter-robot information is unavailable through local sensing, thus reducing the communication requirements. The system is proven to be stable when this algorithm is applied. Results, both using simulated and real quadrotors, show the correct behavior of the algorithm without the use of communications, even when each robot can only sense a subset of the robots in the group.

General Terms

Algorithms, Experimentation, Performance

Keywords

Quadrotors, Formation Control, Limited Field of View

1. INTRODUCTION

The use of Unmanned Aerial Vehicles (UAVs) has been substantially increasing because their design has been simplified and their control techniques made more robust, ultimately enabling for vehicles of smaller size. A Smaller vehicle size allows for a more convenient exploration of multi-UAV systems in confined spaces, and at close range. Quadrotors are typically chosen due to their high maneuverability in such confined spaces. Additionally, through vehicle coordination, the geometry of these systems can be explored to minimize the impact of each vehicle limitations. For example, UAVs can organize to pick and transport heavier objects with a specific shape, or to direct their limited on-board cameras so that their combined images provide full 3D environment coverage. This coordination is achieved through formation control algorithms.

Formation control is a widely studied topic, both in 2D and 3D configurations, with an extensive literature. Formation control laws typically require that each robot knows the relative positions between itself and its neighbors. In most common approaches, the robots obtain this information by sharing via a communication channel their absolute positions with respect to an environment, computed using Global Navigation Satellite Systems (GNSSs) [20] in outdoor environments, Motion Capture Systems (MCS) [19, 1, 9] in indoor environments, or even Montecarlo or SLAM methods [16, 15], if the required on-board positioning sensors and environment conditions allow to do so.

Other approaches use the on-board sensors to directly extract the relative positions between robots, without the need for communication or external systems. However, these sensors have limited capabilities, either on accuracy or Field Of View (FOV). This is especially true for the 3D case, because of the challenging sensing design, either due to the fact that the robot body represents an obstacle for the sensor itself, or because there is a compromise between the sensing area that needs to be covered, and the resolution of the sensor. For example, in [18, 6], on-board cameras extract accurate relative bearing information, which is then used in a formation control algorithm. However, since the cameras are characterized by a limited FOV, the information required from the neighbors that are not directly observed has to be provided also through communication. In [17], the proposed framework uses both range and bearing information collected by each quadrotor to reduce the amount of required observed neighbors per quadrotor. However, if the required neighbors are not observed, communication is still needed to acquire the missing information.

Since high control rates are necessary to stabilize the highly dynamical system formed by these vehicles, all previous approaches become fragile to packet loss or latency in communication links. This is particularly important for short range inter-robot interactions, requiring faster reaction times. To keep communication to a minimum, the sensor constraints must be directly addressed. This was done mostly for ground vehicles, and for sensor range constraints, using potential

*This work has been partially supported by ISR/LARSyS Strategic Funds from FCT [UID/EEA/5009/2013] and FCT SFRH/BD/51928/2012.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

field algorithms that include specialized terms to guarantee that neighbor behavior will not compromise this type of constraint [8, 12]. FOV constraints have also been considered, but just for leader-follower formations, where each follower controls its FOV in a way to maximize the observation of its leader [11, 10, 21]. However, since only one neighbor is considered inside the sensor FOV, the multi-robot system is bounded to a limited number of inter-robot connections, compromising the number of geometries achievable for formation control. Additionally, less connections means reduced system reactivity.

This work tackles these last issues. Sensor range limitations are left aside, and for the experiments, it was made sure that all the necessary inter-robot positions were smaller than the sensor range limits. Also, sensor FOV constraints are considered to be on the horizontal dimension, as typical quadrotor formations simply lie on horizontal planes. With the previous assumptions, the work proposes an algorithm, applicable to quadrotors or any robot with similar dynamics. In particular, we extend the algorithm presented in [5] for a second order holonomic robot dynamics, where each quadrotor uses an additional term to control the critical angle formed between the two neighbors closest to sensor FOV limits. This is performed by making the quadrotor moving closer or further away from the neighbors, depending if the angle needs to increase or decrease respectively. Additionally, this angle is also centered in the FOV to optimize the sensor safety margins.

The algorithm provides two contributions. Firstly, quadrotor trajectories become constrained to cope with the vehicle's sensor limitations when observing multiple neighbors. Secondly, by also controlling angles, this algorithm is able to tackle configurations not achievable by some of the previous algorithms (as the one in [5]) in presence of sensor FOV constraints. This work shows that the terms added in the algorithm do not compromise the system stability. Also, although no discussion is provided about the system initial conditions or disturbances for which the FOV constraints are still maintained, results (simulated and real) show that for typical situations, these constraints are effectively controlled. Thus, the proposed algorithm allows for formation control without the use of inter-robot communication.

This paper is organized as follows. Section 2 provides basic definitions and assumptions about the quadrotor dynamics and its sensing capabilities. Based on those definitions, Section 3 formally introduces the FOV constraints and discusses how to built valid geometric formation configurations. Section 4 presents the formation control algorithm, along with its stability properties. In Section 5, a set of simulation and real experiments are performed to show the correct behavior of the algorithm, and some modifications are designed to cope with real world disturbances. The work concludes with some remarks in Section 6.

2. BASIC DEFINITIONS

This work considers a group of N quadrotors, forming at any time a geometric configuration, defined by a full network graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$. In this graph, $\mathcal{V}$ is the set of N nodes, each one representing a quadrotor, described in a 3D environment by a position $\mathbf{x}_i \in \mathbb{R}^3$, a velocity, $\mathbf{v}_i \in \mathbb{R}^3$, and an attitude, given by the roll ϕ_i, pitch θ_i, and yaw ψ_i, angles. Each quadrotor is considered to have an on-board auto-pilot and inertial sensors, which allows for the control

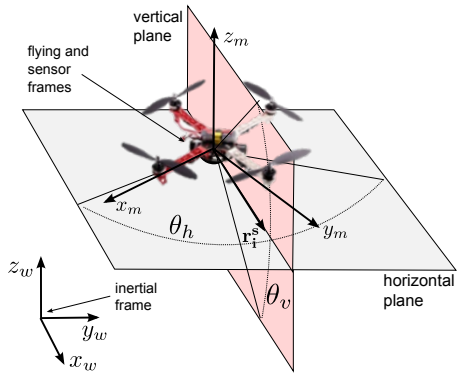

Figure 1: Description of the quadrotor flying frame (x_m, y_m, z_m), with respect to the inertial frame. The quadrotor is placed horizontally for illustrative purposes, but it can have non-zero roll and pitch. Note the sensor horizontal (θ_h) and vertical (θ_v) FOV.

of the vehicle thrust together with the estimation and control of its attitude. This enables the quadrotor to move to any desired 3D position, so it can be considered has an almost perfectly holonomic vehicle. Note that only roll and pitch control together with the thrust control are needed to move the vehicle on the three position axes, leaving the yaw control free to be used for independent purposes (as it will be discussed later). Therefore, we split the pose control of the quadrotor into 3D position control, with a double integrator model, and yaw control with a single integrator dynamics:

$$\ddot{\mathbf{x}}_i = \mathbf{u}_i, \qquad \dot{\psi}_i = u_i^{\psi}, \tag{1}$$

where $\mathbf{u} = (u_i^x, u_i^y, u_i^z)$ and u_i^{ψ} are the desired control inputs for the quadrotor 3D position and yaw angle respectively. The single integrator on the yaw angle is a feature commonly provided by the on-board auto-pilots of quadrotors.

The set of all $N(N-1)/2$ edges of $\mathcal{G}$ is represented by $\mathcal{E}$, where each edge represents relative kinematic information between two quadrotors, namely relative position, $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i = (x_{ij}, y_{ij}, z_{ij})^T$ and, as derived quantity, relative velocity $\mathbf{v}_{ij} = \mathbf{v}_j - \mathbf{v}_i = (v_{ij}^x, v_{ij}^y, v_{ij}^z)^T$. To control $\mathcal{G}$, the quadrotors need to collect information represented by a subset of the previous edges, using their on-board sensors. The information that can be obtained at any time is described by a sensing graph, defined by $\mathcal{G}_S := (\mathcal{V}, \mathcal{E}_S)$, with $\mathcal{E}_S$ representing the set of edges of $\mathcal{G}$, for which the relative positions between the respective quadrotors can be measured using the on-board sensors.

Each quadrotor i is considered to have an on-board sensor, measuring the position $\mathbf{x}_{ij}$ and the velocity $\mathbf{v}_{ij}$, in a reference frame described in Fig. 1. The frame origin is the same as the quadrotors, and the z-axis is aligned with the inertial frame (vertical). In the remainder of this paper, we will call this frame the *flying frame*; it has an attitude of $(\phi, \theta, \psi) = (0, 0, \psi_i)$ with respect to the inertial frame. The sensor has its FOV center in the horizontal xy-plane, considered to be the sensor direction, $\mathbf{r_i^s}$. The FOV is represented by its horizontal, θ_h, and vertical, θ_v, components, defined with respect to the previous described horizontal plane, and a vertical plane formed by $\mathbf{r_i^s}$ and the z-axis of the flying frame. This can be assumed as quadrotors usually move using low roll and pitch values, and the distances between the center of the robot and the sensor are small. In case the

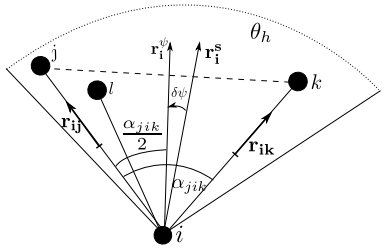

Figure 2: Inter-edge aperture between the edges connecting nodes j and k, to node i, both within the FOV of node i. Apertures formed by different node pairs, for example using node l, are contained inside it. Note the optimal sensor direction, $\mathbf{r_i^\psi}$, equally dividing the occupied area on both sides of the FOV.

quadrotor and the sensors are tilted, the measures extracted on the sensor frame can be transformed into the flying frame, using the ϕ_i and θ_i values acquired from the inertial sensors.

Since the z-axis of the flying frame is the same for all quadrotors, it is useful to decouple the formation control into horizontal and vertical components (as explained later in Section 4). For this reason, the previous relative kinematic information is divided into horizontal components, $\mathbf{x_{ij}^h} = (x_{ij}, y_{ij})^T$ and $\mathbf{v_{ij}^h} = (v_{ij}^x, v_{ij}^y)^T$, and vertical components, z_{ij} and v_{ij}^z. The horizontal position component can be transformed into polar coordinates $(e_{ij}^h, \mathbf{r_{ij}^h})^T$, where e_{ij}^h is the absolute distance between quadrotors i and j, and $\mathbf{r_{ij}^h}$ is a unitary vector defining the direction between i and j. The next section formally introduces the FOV constraints for each quadrotor, and discusses the type of geometric configurations considered in this work.

3. FIELD OF VIEW CONSTRAINTS

The sensor FOV constraints can be mathematically described through the concept of inter-edge aperture. Given two nodes j and k connected to node i, the inter-edge aperture of the respective connection edges, α_{jik}, is defined by the angle between the relative position vectors represented by those edges, as shown in Fig. 2. This concept is divided into horizontal and vertical components, by projecting the edges in the respective planes defined in Fig. 1. Therefore, the inter-edge aperture of every pair of edges belonging to $\mathcal{G}_S$ for a given quadrotor i must be smaller than θ_h in the horizontal case, or θ_v in the vertical case. Only the biggest aperture for each quadrotor needs to be considered, because if this value is smaller than the FOV limit, all the others apertures will be also smaller, as illustrated in Fig. 2. Our algorithm operates on the FOV constraints by directly controlling these apertures, as explained in the next section.

Note that if the FOV constraints are verified, there is always a quadrotor yaw ψ_i, describing the sensor direction, $\mathbf{r_i^s}$, that allows all neighbors to be observed. In fact, it is possible to define an optimal sensor direction r_i^ψ, illustrated in Fig. 2, defined so that it equally distributes the sensing area around the center of the largest inter-edge aperture, optimizing the measurement safety margins.

A targeted geometric configuration is defined by a graph $\mathcal{G}$ whose edges correspond to desired inter-robot positions $\mathbf{x}_{ij}^d = (x_{ij}^d, y_{ij}^d, z_{ij}^d)$, and zero velocity. From this information, all the desired apertures between sets of three nodes, α_{jik}^d can be obtained. To control this configuration, a formation control graph $\mathcal{G}_F := (\mathcal{V}, \mathcal{E}_F)$ is defined, where each edge in $\mathcal{E}_F$ corresponds to a controlled inter-robot distance. The configuration is controllable if the following two conditions are verified: $\mathcal{G}_F$ is rigid, so that it has a unique solution (the reader is referred to [14] for more details about graph rigidity); $\mathcal{G}_S$ must contain enough information to enable the control of all edges in $\mathcal{G}_F$. This work assumes $\mathcal{G}_S$ and $\mathcal{G}_F$ fixed and defined a priori. Note that, in order to achieve the desired $\mathcal{G}_S$, the robots need to start the algorithm in positions that are already respecting the FOV constraints.

Note that the previous section described the sensor direction, $\mathbf{r_i^s}$ to be always directed and controlled horizontally. Therefore, this work assumes geometric configurations where all quadrotors have the same height, or smaller than the vertical FOV, avoiding therefore vertical FOV violations. Additionally, this work considers the edges in $\mathcal{E}_S$ and $\mathcal{E}_F$ to be bi-directional. For $\mathcal{E}_S$ edges, this concretely means that the quadrotors forming any given edge can mutually sense each other. For $\mathcal{E}_F$ edges, this concretely means that both quadrotors forming the edge actively participate on the control of the mutual distance. These bi-directionality assumptions are made to reduce the distance instability between quadrotors, caused by possible delays in the vehicle perception-to-action loop.

The previous assumptions can limit the formation configurations that can be considered, as the number of robots increases. This work focuses on the simplest type of formations, placing the robots on the convex hull of the target formation shape. Robots can also be placed inside the convex hull, at different heights to avoid sensor occlusion. Their neighbors can be selected randomly as long as the largest inter-neighbor aperture respects the sensor FOV constraints, as explained above. However, should the previous edge bi-directionality assumptions be kept, since robots inside the convex hull observe a reduced neighbor set, the number of inter-robot connections achievable by the other robots is also reduced. Uni-directional edges can be used if the overall formation rigidity is kept, but this case is left for future work. The next section introduces the formation control algorithm, and how it provides sufficient controllability to establish $\mathcal{G}_F$.

4. CONTROL ALGORITHM

The algorithm leverages two distinct components: the control of the inter-robot distance and the maintenance of FOV visibility. To control the inter-robot distances, each distance is given a weight, describing how strongly the quadrotor controls it. The weighting information can be elegantly represented using a Laplacian matrix, $\mathbf{L}$, where the element $\mathbf{L_{ij}}$ defines the control weight for the distance between neighbors i and j. This matrix is positive definite, the sum of the elements of each line sum to zero, and $\mathbf{L_{ii}} = -\sum_{j \neq i} \mathbf{L_{ij}}$. If $\mathbf{L_{ij}} = 0$, the distance between quadrotors i and j is not directly controlled. In this work, $\mathbf{L}$ is constant, since $\mathcal{G}_F$ is assumed constant. Each quadrotor controls distances using inter-robot position information measured by its on-board sensors, in the flying frame. Since the z-axis of this frame is the same for all quadrotors, and equal to the z-axis of the inertial frame, this dimension is controlled with a simple consensus equation, similarly to [3, 13], as follows:

$$u_i^z = k_p \sum_{j=1}^{N} \mathbf{L_{ij}} \left(z_{ij}^d - z_{ij} \right) + k_v \sum_{j=1}^{N} \mathbf{L_{ij}} v_{ij}^z, \qquad (2)$$

where u_i^z is the desired vertical acceleration for quadrotor i, k_p and k_v are gain parameters for the position and the velocity components of the controller. To control the horizontal dimension, the range and bearing controller proposed in [5] is used, but extended for a double integrator case and simplified to the case of holonomic vehicles:

$$\mathbf{u_i^h} = k_p \sum_{j=1}^{N} \mathbf{r_{ij}^h L_{ij}} \left(e_{ij}^h - e_{ij}^d \right) + k_v \sum_{j=1}^{N} \mathbf{L_{ij} v_{ij}^h}, \qquad (3)$$

where $\mathbf{u_i^h} = (u_i^x, u_i^y)^T$ is the desired horizontal input in the flying frame for quadrotor i and e_{ij}^d is the desired horizontal distance between quadrotors i and j, obtained from $||(x_{ij}^d, y_{ij}^d)||$. In both the previous equations, the position component aims at achieving the formation requirements, while the velocity component stabilizes the second order dynamic system. Distances controlled in this way allow the edges representing them to be added into $\mathcal{G}_F$. To control these distances, the respective relative position information is needed, which means these control edges must also belong to $\mathcal{E}_S$.

The second component of the algorithm controls the FOV constraints, which can happen as the formation rotates around its z axis, or when quadrotors move to their desired positions. To solve this issue, two new terms are included, allowing each quadrotor to additionally control the largest observed inter-edge aperture, and the sensor direction, $\mathbf{r_i^s}$, in the horizontal plane. A simple yaw controller is used to derive $\mathbf{r_i^s}$ to the optimal direction defined in the previous section, $\mathbf{r_i^\psi}$. This is done by computing the angle between the two vectors, $\delta\psi$, illustrated in Fig. 2. Note that $\mathbf{r_i^s}$ is always known, and $\mathbf{r_i^\psi}$ is computed by averaging $\mathbf{r_{ij}}$ and $\mathbf{r_{ik}}$. The obtained $\delta\psi$ is given as the control input of the yaw controller and driven to zero:

$$u_i^\psi = -k_\psi \delta\psi_i, \qquad (4)$$

where u_i^ψ is the yaw control input defined in Eq. 1, and k_ψ is a control gain. The biggest inter-edge aperture observed by quadrotor i, α_{kij}, is controlled by adding a term to Eq. 3, as follows:

$$\mathbf{u_i^{h*}} = \mathbf{u_i^h} + k_\alpha(\alpha_{kij}^d - \alpha_{kij})(K\mathbf{r_i^c} + K_\perp \mathbf{r_i^{c\perp}}), \qquad (5)$$

where α_{kij}^d is the desired aperture between quadrotor i and neighbors j and k currently defining the biggest aperture, k_α is a control gain, and K and $K_\perp$ will be chosen according to the stability analysis presented in Proposition 1. Vector $\mathbf{r_i^c}$, illustrated in Figs. 3 and 4, is a unitary vector defining the direction between quadrotor i and the averaged formation center, $\mathbf{C_i}$. This center is defined for each quadrotor, and its displacement from the robot is computed using the first term of Eq. 3:

$$\mathbf{r_i^c} e_i = \sum_{j=1}^{N} \mathbf{r_{ij}^h L_{ij}} e_{ij}^h, \qquad (6)$$

where e_i is the distance between quadrotor i and $\mathbf{C_i}$. The distance between a neighbor j and $\mathbf{C_i}$ is defined as e_i^j. Note that $\mathbf{r_i^c}$ is always in between the neighbors closer to the FOV edges. Therefore, if α_{kij} is too large, robot i generates a force pointing backwards, in a direction that will always decrease α_{kij}. The contrary occurs when α_{kij} is too small. This corresponds to a direct control of the sensor FOV constraint during the algorithm's operation.

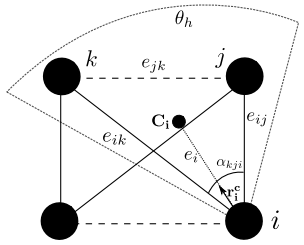

Figure 3: Possible $\mathcal{G}_F$ for a square formation and an horizontal FOV of less than 90°. Filled edges correspond to direct distance control. Dashed edges are included if aperture control is activated. Note quadrotor i's weighted formation center.

With the additional control terms, each quadrotor also indirectly controls the distances between the neighbors themselves, without them knowing the respective relative position information. Consider the example in Fig. 3, when the FOV is less than 90°. The most complete $\mathcal{G}_S$ possible with these constraints is the one shown in the figure (excluding the dashed lines). So, with just direct distance control, no rigid graph can be defined. However, if quadrotor i additionaly controls α_{jik}, one can see that e_{jk} is fully expressed in terms of the direct controlled quantities, e_{ij}, e_{ik}, and α_{jik}. So, the edge between neighbors j and k is automatically included in $\mathcal{E}_F$, which does not need to belong to $\mathcal{G}_S$. Each quadrotor can control an additional edge in this way, allowing the establishment of rigid, or even fully connected, formation graphs that could not possibly be formed before.

The next proposition shows that the new proposed terms do not compromise the system stability. The algorithm can be easily extended to control more than one inter-edge aperture for each quadrotor, but the presented stability properties are related to the largest. Additionally, it is not clear that these properties hold if the neighbors forming the aperture changes in time. Since the previous case is rare, in this paper we focus on characterizing the algorithm behavior when the inter-edge aperture remains the same during operation, and leave the previous issue for future work.

PROPOSITION 1. *As long the necessary FOV constraints are not violated, the multi-robot system, with each quadrotor i described by the dynamics in Eq. (1) and applying the controller presented in Eqs. (2), (3), (4), and (5), is stable, for any chosen set of weights described in $\mathbf{L}$, and any k_p, k_v, K and $K^\perp$ greater than zero.*

PROOF. Note that [13] already proved convergence properties for vertical controller components Eq. (2). Also,the component in Eq. (4) is independent from the actual formation control. Therefore, if the formation converges, $\mathbf{r_i^\psi}$ converges as well, allowing the yaw controller to stabilize $\delta\psi_i$ to zero, since it is a proportional control applied to a single integrator system.

The proof for the horizontal controller components of $\mathbf{u_i^{h*}}$ follows the reasoning of [5], which performs the analysis separately for each robot, and its neighbors are assumed to have fixed positions, and then combines the results at the end. To simplify the proof, the bias terms (α_{kij}^d and e_{ij}^d) are set to zero, but the stability still holds for non-zero terms (such terms would only change the equilibrium point).

Let us assume the case for quadrotor i, described in Fig. 4, representative of the horizontal control components. Quadro-

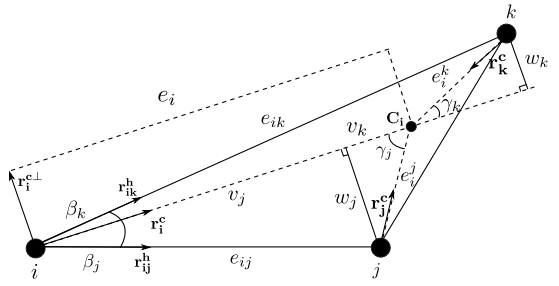

Figure 4: Horizontal formation control formulation.

tors j and k correspond to quadrotor i neighbors forming the largest aperture in its FOV. The position of quadrotor i's formation center, $\mathbf{C_i}$, can be expressed in the inertial frame as $\mathbf{x_i} + \mathbf{r_i^c} e_i$, where the second term comes from Eq. 6. This expression does not depend on $\mathbf{x_i}$ if $\sum_{j \neq i} \mathbf{L_{ij}} = 1$, which means $\mathbf{C_i}$ doesn't change with quadrotor i's movements. If the neighbors are assumed to have fixed positions, their distances to $\mathbf{C_i}$, e_i^j and e_i^k, are constant. The axes of quadrotor i's reference frame are changed to $\mathbf{r_i^c}$ and $\mathbf{r_i^{c\perp}}$, representing respectively the radial and orthogonal axis with respect to $\mathbf{C_i}$. The velocity of $\mathbf{C_i}$ in this new frame is decomposed on the radial, $\dot{e}_i$ and orthogonal, $\dot{e}_i^\perp$, axes. Note that this velocity corresponds to the second term of Eq. 3, and that $\dot{e}_i$ is the velocity of e_i, representing the distance between quadrotor i and $\mathbf{C_i}$. From the previous definitions, a simple Lyapunov function is chosen to analyze the stability of the system for quadrotor i:

$$V_i(\alpha_{kij}, e_i, \dot{e}_i, \dot{e}_i^\perp) = \frac{1}{2}(k_\alpha \alpha_{kij}^2 + k_p e_i^2 + (\dot{e}_i)^2 + (\dot{e}_i^\perp)^2),$$

which is greater than zero except in $V_i(0,0,0,0)$. The four components were considered because they represent the states that are being controlled (aperture, distance, and radial and orthogonal velocity). The derivative of V_i with respect to time can be expressed as:

$$\dot{V}_i = k_\alpha \alpha_{kij} \dot{\alpha}_{kij} + k_p e_i \dot{e}_i + \dot{e}_i \ddot{e}_i + \dot{e}_i^\perp \ddot{e}_i^\perp,$$

where $\ddot{e}_i$ and $\ddot{e}_i^\perp$ can be expressed from the $\mathbf{u_i^{h*}}$ terms in Eqs. 3 and 5, projected into the radial and orthogonal components respectively:

$$\ddot{e}_i = K k_\alpha \alpha_{kij} - k_p e_i - k_v \dot{e}_i,$$
$$\ddot{e}_i^\perp = K_\perp k_\alpha \alpha_{kij} - k_v \dot{e}_i^\perp.$$

Therefore, $\dot{V}_i$ can be simplified, by removing the equal terms in its expression, to:

$$\dot{V}_i = k_\alpha \alpha_{kij} \dot{\alpha}_{kij} + k_\alpha \alpha_{kij}(\dot{e}_i K + \dot{e}_i^\perp K_\perp) - k_v(\dot{e}_i)^2 - k_v(\dot{e}_i^\perp)^2.$$

The last two terms are always negative, which leaves the study of the first component. From Fig. 4, the aperture α_{kij} can be divided into $\beta_j + \beta_k$, where $\beta_{(*)} = \arctan(w_{(*)}/v_{(*)})$, and $(*)$ is either j or k. By differentiating $\beta_{(*)}$, $\dot{\alpha}_{kij}$ can be defined as:

$$\dot{\alpha}_{kij} = \frac{v_j \dot{w}_j - \dot{v}_j w_j}{e_{ij}^2} + \frac{v_k \dot{w}_k - \dot{v}_k w_k}{e_{ik}^2}.$$

From the figure, on can define $v_{(*)} = e_i - e_i^{(*)} \cos(\gamma_{(*)})$ and $w_{(*)} = e_i^{(*)} \sin(\gamma_{(*)})$, where $\gamma_{(*)}$ is the angle going from $\mathbf{r_{(*)}^c}$ to $\mathbf{r_i^c}$. Recalling that neighbors have fixed positions with respect to $\mathbf{C_i}$, $\gamma_{(*)}$ only depends on $\dot{e}_i^\perp$, according to the linear-to-angular velocity equation, $\dot{e}_i^\perp = \dot{\gamma}_{(*)} e_i$. Using the previous result for $\dot{\gamma}_{(*)}$, and recalling that $e_i^{(*)}$ is constant, the derivatives of the previous expressions for $v_{(*)}$ and $w_{(*)}$ are as follows:

$$\dot{v}_{(*)} = \dot{e}_i + \dot{e}_i^\perp \sin(\gamma_{(*)}) \frac{e_i^{(*)}}{e_i}, \quad \dot{w}_{(*)} = \dot{e}_i^\perp cos(\gamma_{(*)}) \frac{e_i^{(*)}}{e_i}.$$

From the previous result, and noting that the alternative definitions $w_{(*)} = e_{i(*)} \sin(\beta_{(*)})$ and $v_{(*)} = e_{(*)} \cos(\beta_{(*)})$, the previous expression for $\dot{\alpha}_{kij}$ can be re-arranged, to isolate the terms in $\dot{e}_i^\perp$ and $\dot{e}_i$, as follows:

$$\dot{\alpha}_{kij} = \dot{e}_i^\perp \left(\frac{e_j}{e_i} \frac{\cos(\gamma_j + \beta_j)}{e_{ij}} + \frac{e_k}{e_i} \frac{\cos(\gamma_k + \beta_k)}{e_i e_{ik}} \right) - \dot{e}_i \left(\frac{\sin(\beta_j)}{e_{ij}} + \frac{\sin(\beta_k)}{e_{ik}} \right).$$

One can now choose K and $K_\perp$ of the aperture controller to eliminate the previous components, ending with the following result:

$$K = \left(\frac{\sin(\beta_j)}{e_{ij}} + \frac{\sin(\beta_k)}{e_{ik}} \right)$$
$$K_\perp = -\left(\frac{e_j}{e_i} \frac{\cos(\gamma_j + \beta_j)}{e_{ij}} + \frac{e_k}{e_i} \frac{\cos(\gamma_k + \beta_k)}{e_{ik}} \right)$$
$$\dot{V}_i = -k_v(\dot{e}_i)^2 - k_v(\dot{e}_i^\perp)^2.$$

Note that $\dot{V}_i \leq 0$, and therefore using the Lyapunov theorem, the system with the proposed controller is stable and converges to a subset of the state-space defined by $\dot{V}_i = 0$. This can be extended to all quadrotors, by setting $V = V_1 + ... + V_N$, where V_i is the previous Lyapunov function but for each quadrotor. $V > 0$ except in $V(0) = 0$, and $\dot{V} <= 0$, and therefore, the system as a whole is also stable. Finally, note that $k_p \sum_{j \neq i} \mathbf{L_{ij}}$ can always be transformed into $k_p^* \sum_{j \neq i} \mathbf{L_{ij}^*}$ for each quadrotor, where $\sum_{j \neq i} \mathbf{L_{ij}^*} = 1$, necessary to guarantee that $\mathbf{C_i}$ does not depend on robot i's movement. This allows the use of any Laplacian matrix in this system, regardless of k_p. □

The system is stable, but it converges to the set described as $\dot{V} \leq 0$, which, from the previous proof, only guarantees that the quadrotor velocities are zero. Deadlocks can occur, especially if the configuration is ill defined, i.e. the set of desired ranges and apertures correspond to an impossible configuration. In this case, the system will converge to a situation where the aperture controller will counter-act the distance controller, creating the deadlock. The study of deadlocks is considered to be future work. However, note that those already existed in [5], referenced as local minima.

Finally, the gains K and $K_\perp$ found for the aperture controller are analyzed. The value of K is related to the controller radial component. Its value is intuitive, saying that it is always bigger than zero, as $\beta_{(i)} \leq \pi$, and it is bigger as the angle increases to $\pi/2$, corresponding to the point of maximum influence of the controller in the angle. Also, as e_{ij} decreases, the gain increases since the influence on the aperture also increases. The value of $K_\perp$ is related to the controller orthogonal component, and it is less intuitive. However, note that it uses cosine instead of sine functions, indicating that it is controlling an axis orthogonal to the one K controls. For example, if all quadrotors are found in a line, $K = 0$, since moving on the line does not control the aperture, but $K_\perp \neq 0$, since moving orthogonally to the line increases the aperture. For simplicity, and due to time constraints, this work considers $K_\perp = 0$, but the experiments show that the system can converge without this component.

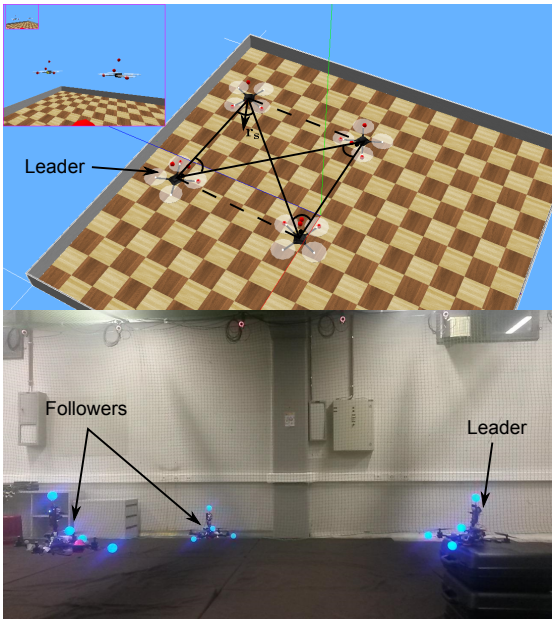

Figure 5: Experimental setups. (a) Webots simulator using a square formation - note that only two quadrotors are visible from the simulated camera sensor view (upper left corner). (b) Triangle formation in the flying arena endowed with a MCS, manufactured by Motion Analysis Inc. and composed of 20 Osprey cameras able to track the quadrotors using a set of reflective markers in an useful volume of 4x7x2.5 m. Note the static leader on the right.

Future work will include experiments done with a non-zero value of $K_\perp$ in the controller.

5. EXPERIMENTS

The algorithm was tested through a set of experiments conducted using the platforms described in Fig. 5. Both simulation (Fig. 5 (a)) and real (Fig. 5 (b)) environments were considered. In reality, Hummingbird quadrotors, manufactured by Ascending Technologies[1], were used. Each quadrotor was equipped with a set of active markers. An on-board camera with a 320x240 pixel frame and an horizontal FOV of 90° was used to detect the active markers on the neighbors, and extract their relative positions using an algorithm similar to the one presented in [4]. After calibration, the sensor noise had a standard deviation of 13 cm at a 3 meter range. The high-fidelity robotic simulator Webots[2] was used as the simulating environment, with a standard quadrotor model, not tuned to match the real quadrotors. The markers were simulated as colored blobs, and an ideal camera sensor was simulated with the same FOV. Perturbations were added on the simulated images to achieve the same noise levels measured in reality. Although in reality sensor noise decreases at smaller distances, in simulation it was kept with the same intensity at all distances. Finally, the actuation noise in reality was not measured, and in simulation it was set to be zero. A positioning ground truth is provided in reality through the MCS presented in Fig. 5. Additional details about the real setup and the system composed of the camera and active markers can be found in [2].

[1]http://www.asctec.de/
[2]https://www.cyberbotics.com/

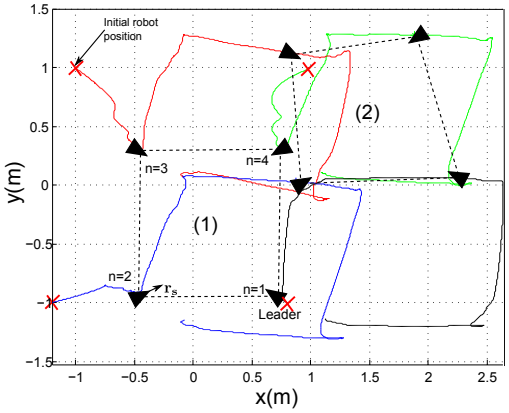

Figure 6: Evolution of the quadrotor trajectories performed in simulation using the formation control algorithm and the defined leader goal following strategy. Each robot is represented by a black triangle, and each color illustrates a trajectory of a specific quadrotor. The triangle orientation is the same as $\mathbf{r_i^s}$. (1) and (2) represent snapshots, when the formation is respectively stable and distorted.

The formation task was performed using four and three quadrotors, in simulation and real environments, respectively. Although not needed for the formation control, a leader was selected with the objective of anchoring the formation in the environment, and for the simulation experiments, to move the formation. The simulation experiments use a square formation configuration, depicted in Fig. 5 (a), and also described in Fig. 3, to assess rigidity issues. The square sides measured 1.2 m, and each quadrotor observed other two quadrotors, controlling an aperture of 45° between them. The formation leader was chosen at random. The real experiments use a triangle formation configuration, depicted in Fig. 5 (b), where the leader was static in the environment. The desired distances were 1.5 m between the leader and the followers, and 2.12 m between the followers. The desired aperture for each follower was also 45°. Both graphs, if the aperture control is included, are fully connected and rigid. The desired relative heights between the quadrotors in both scenarios were set to zero.

To move the leader in simulation, an extra term is added to the vertical and horizontal formation control laws in Eqs. (5) and (2), as follows:

$$u_i^{hl} = k_f u_i^{h*} + k_g k_v(\mathbf{v_i^h} - \mathbf{v^d}),$$
$$u_i^{zl} = k_f u_i^z + k_g k_v(v_i^z - v_i^{zd}),$$

where k_f and k_g regulate the importance of formation cohesion and goal following behaviors, both set to 0.5, indicating equal importance. The values of k_p, k_v and k_α were set to 1.6, 2.2 and 1.0 respectively. Therefore, the leader performs both goal following and formation cohesion, similar to the work presented in [7], and followers perform just the formation cohesion. The Laplacian weights between the leader and the followers were set to 1, as opposed to the weights between the followers, always chosen to be $\frac{1}{N_i}$, where N_i is the number of neighbors of robot i. This is because, in contrast to the leader, follower movements do not translate directly

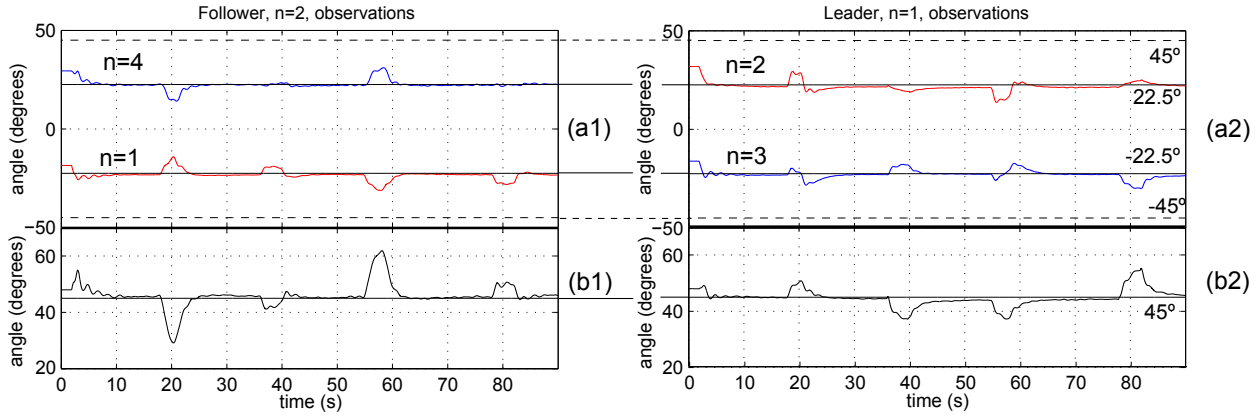

Figure 7: Controlled bearings (a) and aperture (b) values, observed from quadrotors n=2, and n=1. Desired bearing and apertures depicted using horizontal lines, with respected values shown on the right. The neighbor to which each bearing line corresponds, is indicated in text on the left. FOV limits shown with dashed lines.

the behavior that is desired for the robot, but rather a response to other neighbor changes, that can either have lag, noise, or be wrong. Therefore, these behaviors should be averaged. The next sections describe separately the results obtained in each environment.

5.1 Simulation Results

The formation stability, rigidity, and correct maintenance of the sensor FOV constraint were tested by moving the square formation through leader commands with a certain velocity in the x and y dimensions separately, as shown in Fig. 6 (the z velocity is set to zero). The used velocity value was of 0.5 m/s. The initial quadrotor positions were set so that all necessary FOV constraints were initially met. The proposed algorithm was then activated, so that the quadrotors could converge to the desired configuration. For one experimental run, Fig. 8 shows the progress of all the horizontal distance errors between two pairs of quadrotors, and Fig. 7 shows the neighbor bearings and apertures observed by the leader $n = 1$ and the follower $n = 2$ (numbers described in Fig. 6). The bearings are defined with respect to the camera FOV center, $\mathbf{r_i^s}$, illustrated in Fig. 5 (a).

Fig. 8 shows that the formation always converges to the right configuration, even after the perturbation caused by leader movements, showing its rigidity and stability. Note that, when the leader moves, the configuration is distorted (also observed in Fig. 6), and with higher intensity on the y axis (observed by the higher distance errors). The distortions happen because only the leader has the knowledge of the desired velocity and the followers simply follow the leader. The distortion is bigger on the y axis, because that is the direction where the leader is aligned with the follower $n = 4$, that only relies on other followers for information. This will generate a bigger movement delays between the leader and that follower, creating a bigger distance and aperture errors.

Fig. 7 shows similar perturbations on the bearing and aperture values. However, Fig. 7 (a), shows the convergence of neighbor bearings, in both leader and follower cases, to symmetric values with respect to $\mathbf{r_i^s}$, meaning that the yaw controller described in Eq. (4) optimizes the sensor FOV. Additionally, Fig. 7 (b) confirms, for the leader and follower cases, the convergence of the aperture to the desired values, which in combination with results presented in Fig. 8, shows the correct behavior of the horizontal controller proposed in Eqs. (2), (3), and (5), and the correct maintenance of the FOV constraints. The aperture distortions are also higher for movements on the y axis, for the same reasons as before.

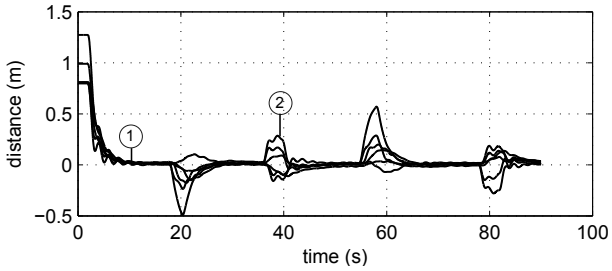

Figure 8: Horizontal distance errors between two pairs of quadrotors, during the experiment run corresponding to Fig. 6. The two snapshots described on that figure are highlighted here using circles.

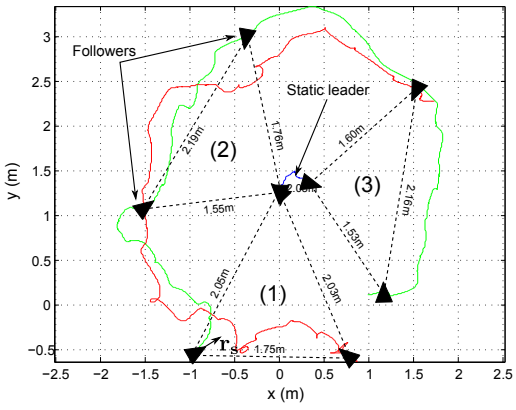

Figure 9: Real quadrotor trajectories tracked by the MCS, during a run of the algorithm. Note that the leader is static. Each color represents one quadrotor. Vector $\mathbf{r_i^s}$ is defined by the orientation of the described triangles. (1), (2), and (3) represent three snapshots of the formation, where (1) corresponds to the initial follower positions.

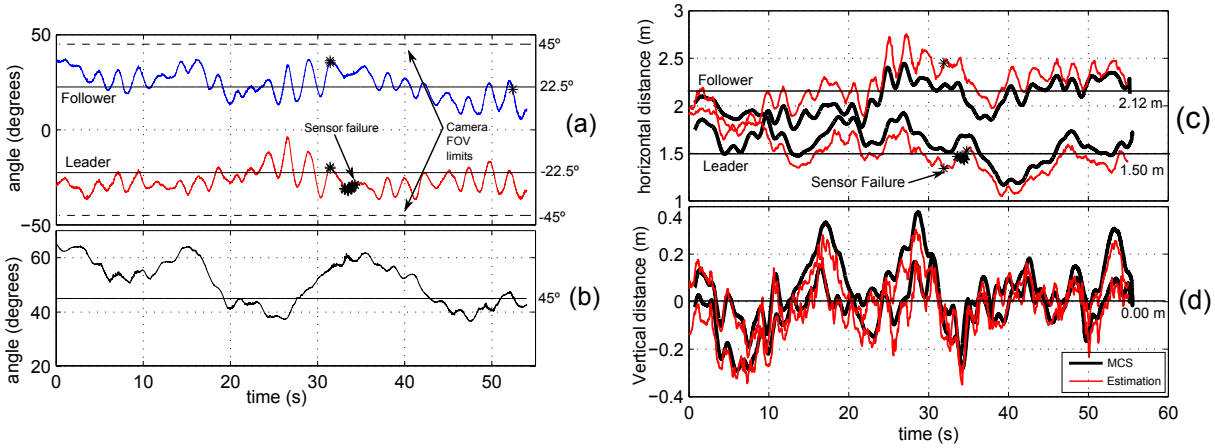

Figure 10: Neighbor bearings (a), aperture (b), and distances observed in the local frame of one of the followers for a real experiment. The distance is divided into horizontal (c) and vertical (d) components. Desired bearings, apertures, and distances are depicted on the respective plot using horizontal lines, with their values written on the right. The neighbor to which each bearing or horizontal distance line corresponds is indicated in text on the left of the line. Vertical ranges of both neighbors indistinguishable since they are both close to zero. The aperture is related to both neighbors. All values are tracked using the MCS, but on the distance plots, the values estimated from the sensor data are also shown. Black dots on the lines signal moments where the follower stopped receiving sensor data of the respective neighbor for more than 100 ms.

5.2 Real Experiments

Similar experiments were performed using the triangle formation described above, using the real setup depicted in Fig. 5 (b). The leader was placed in the environment, and it remains static through all the experiment. The followers were teleoperated into their initial positions using the MCS position feedback, in order to guarantee that their on-board cameras FOV initially cover their neighborhood. The algorithm was then activated, running exclusively on-board, using the camera for sensor feedback, until a stop and land command was broadcast. The resulting trajectories for each quadrotor, from the start of the algorithm to the end of one experiment, are shown in Fig. 9. The neighbor bearing, aperture, and distance information was tracked throughout the experiment for one of the followers, and shown in Fig. 10. For the bearing and aperture, the results show that, although the followers rotate around the leader, their values remain close to the desired, as in the simulated results, showing the correct behavior of the controller. This allowed loss of sensor data to be kept to a minimum, as shown in the figure. Note that, because of follower rotations, the leader had to be also rotated by hand, so that the followers would keep observing the leader.

Follower rotations happen due to biases present on actuation and on sensing. Fig. 10 (c) shows an example of a sensor bias on the horizontal plane, observed by the differences between MCS and on-board estimations of the tracked relative positions. These biases create unwanted forces competing with the formation control algorithm. Since the algorithm forces are tangent to the edge between quadrotors, the algorithm becomes weak on the radial axis, allowing relatively small forces to still be able to generate rotation movements. The biases can be different for each quadrotor, it could happen that they generate rotations in different directions, that would lead to a steady increase of the follower aperture observed by the leader. In this case, the leader would perform aperture control, going backwards to maintain the desired value. If these biases are constant, the system would move backwards until the end of the experiment. However, these biases can be minimized using integrators, included on the range control on each edge. Note that more than one edge is needed to compensate the bias, as range control on an edge is tangential to that edge. An integrator was also included in the height controller. This can generate low frequency oscillations around the desired relative displacements, which is possible to observe in Fig. 10, in both vertical, with a maximum of 40 cm, and horizontal components, with a maximum of 30 cm. However, despite those oscillations, the algorithm was still successfully maintaining the desired formation.

6. CONCLUSIONS

This work proposed a distributed formation control algorithm for quadrotors, that actively controls the system to satisfy the FOV constraints of the robot on-board sensors, for multiple observed neighbors. This allows for more complex and dynamic formation configurations with minimal or no communication requirements. Results show that the system is stable and that the FOV constraints are satisfied in both simulation and reality.

As future work, the previously discussed deadlocks will be investigated, together with the inclusion of goal following control for formation guidance. These experiments are to be performed with the orthogonal component of the aperture control activated. Furthermore, the sensing and control edge bi-directionality assumptions will be relaxed in order to consider additional formation configurations, paying special attention to how this might impact the formation stability. The results reported in this paper also outline that it is relevant to reproduce, in simulation, the real sensing and actuation mismatches, to further study the bias effects. This algorithm can be strengthened if the stability properties are found for the control of multiple apertures, and if the initial conditions that guarantee the FOV maintenance during the system operation are investigated.

REFERENCES

[1] F. Augugliaro, A. P. Schoellig, and R. D'Andrea. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In *International Conference on Intelligent Robots and Systems*, pages 1917–1922, 2012.

[2] D. Dias, R. Ventura, P. U. Lima, and A. Martinoli. On-board vision-based 3d relative localization system for multiple quadrotors. In *International Conference on Robotics and Automation*, 2016 (to appear).

[3] M. Egerestedt. Graph-theoretic methods for multi-agent coordination. In *ROBOMAT*, 2007.

[4] M. Faessler, E. Mueggler, K. Schwabe, and D. Scaramuzza. A monocular pose estimation system based on infrared leds. In *IEEE International Conference on Robotics and Automation*, 2014.

[5] R. Falconi, S. Gowal, and A. Martinoli. Graph-based distributed control of non-holonomic vehicles endowed with local positioning information engaged in escorting missions. In *International Conference on Robotics and Automation*, pages 3207–3214, 2010.

[6] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. Bülthoff, and P. Giordano. Modeling and control of uav bearing formations with bilateral high-level steering. *International Journal of Robotics Research*, 31(12):1504–1525, 2012.

[7] D. Goldin and J. Raisch. Controllability of second order leader-follower systems. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 223–238, 2010.

[8] M. Ji and M. Egerstedt. Connectedness preserving distributed coordination control over dynamic graphs. In *American Control Conference*, pages 93–98, 2005.

[9] N. M. M. Turpin and V. Kumar. Decentralized formation control with variable shapes for aerial robots. In *IEEE International Conference on Robotics and Automation*, pages 23–30, 2012.

[10] D. Panagou and V. Kumar. Maintaining visibility for leader-follower formations in obstacle environments. In *IEEE International Conference on Robotics and Automation*, pages 1811–1816, 2012.

[11] D. Panagou and K. Kyriakopoulos. Cooperative formation control of underactuated marine vehicles for target surveillance under sensing and communication constraints. In *IEEE International Conference on Robotics and Automation*, pages 1871–1876, 2013.

[12] G. Pereira, A. K. Das, V. Kumar, and M. Campos. Formation control with configuration space constraints. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2755 – 2760, 2003.

[13] W. Ren and E. Atkins. Second-order consensus protocols in multiple vehicle systems with local interactions. In *Guidance, Navigation, and Control Conference*, 2005.

[14] R. O. Saber and R. M. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. In *IEEE International Conference on Decision and Control*, pages 2965–2971, 2002.

[15] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. Achtelik, M. Chl, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, H. Gim, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, and L. Meier. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments. *IEEE Robotics & Automation Magazine*, 21(3):26–40, 2014.

[16] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar. Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In *Robotics: Science and Systems*, 2013.

[17] G. Stacey, R. Mahony, and P. Corke. A bondgraph approach to formation control using relative state measurements. In *European Control Conference*, pages 1262–1267, 2013.

[18] R. Tron, J. Thomas, G. Loianno, J. Polin, V. Kumar, , and K. Daniilidis. Vision-based formation control of aerial vehicles. In *Robotics: Science and Systems*, 2014.

[19] M. Turpin, N. Michael, and V. Kumar. Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, 33(1-2):143–156, 2012.

[20] G. Vásárhelyi, C. Virágh, G. Somorjai, N. Tarcai, T. Szorényi, T. Nepusz, and T. Vicsek. Outdoor flocking and formation flight with autonomous aerial robots. In *International Conference on Intelligent Robots and Systems*, pages 3866–3873, 2014.

[21] C. Verginis, C. Bechlioulis, D. Dimarogonas, and K. Kyriakopoulos. Decentralized 2-d control of vehicular platoons under limited visual feedback. In *IEEE International Conference on Intelligent Robots and Systems*, 2015.